

CLAIMS

- 1 1. A graph walking system, comprising:
 - 2 a binding system for binding a graph observer with a graph, for binding node
 - 3 patterns to node observers to generate at least one node pairing, and for binding the graph
 - 4 observer to at least one node pattern-node observer pairing;
 - 5 graph walking logic for systematically walking through nodes within the directed
 - 6 non-cyclic graph;
 - 7 a pattern testing system for determining if an encountered node matches one of
 - 8 the node patterns;
 - 9 an event manager for generating an encountered event when one of the node
 - 10 observers is bound to a matching node pattern; and
 - 11 a pruning system that can deactivate the graph observer with respect to sub-nodes
 - 12 of the encountered node if a bound node observer determines that there is no interest in
 - 13 the sub-nodes.
- 1 2. The graph walking system of claim 1, wherein the encountered event is handled by the
- 2 bound node observer.
- 1 3. The graph walking system of claim 1, wherein the graph walking logic walks through
- 2 the graph in a top down hierachal manner.

1 4. The graph walking system of claim 1, wherein the pruning system can reactivate a
2 deactivated graph observer after the sub-nodes of the encountered node have been
3 walked.

1 5. The graph walking system of claim 1, wherein the event manager generates a
2 completed event for each node observer that received an encountered event and that did
3 not cause the graph observer to become deactivated.

1 6. The graph walking system of claim 5, wherein the completed event can cause the
2 graph walking logic to repeat the walk through the sub-nodes.

1 7. The graph walking system of claim 1, wherein the pruning system can further cause
2 the graph walking logic to bypass walking of the sub-nodes if the graph observer has
3 been deactivated and no other active graph observers exist.

1 8. A system for analyzing a graph of hierarchical data, comprising:

2 a system for binding a plurality of graph observers to a graph, wherein each graph

3 observer is further bound to a set of node patterns and a set of node observers;

4 graph walking logic for systematically walking through nodes within the graph;

5 a first pruning system that can be instructed by a node observer bound with an

6 associated graph observer to deactivate the associated graph observer until a set of sub-

7 nodes for the encountered node has been walked; and

8 a second pruning system that can instruct the graph walking logic not to walk the

9 set of sub-nodes for the encountered node.

1 9. The system of claim 8, wherein the second pruning system will cause the set of sub-

2 nodes not to be walked only if all of the plurality of graph observers have been

3 deactivated.

1 10. The system of claim 8, further comprising a pattern testing system for determining if

2 the encountered node matches one of the node patterns.

1 11. The system of claim 8, further comprising an event manager for generating an

2 encountered event when one of the node observers is bound to a matching node pattern.

1 12. A method for analyzing a graph of hierarchical data, comprising the steps of:
2 binding a plurality of graph observers to a graph, wherein each graph observer is
3 further bound to a set of node patterns and a set of node observers;
4 systematically walking through nodes within the graph;
5 generating an encounter event and handling the encounter event with a bound
6 node observer when one of the node patterns matches an encountered node;
7 deactivating the graph observer associated with the bound node observer if the
8 bound node observer determines that a set of sub-nodes of the encountered node should
9 be pruned; and
10 bypassing the walking of the set of sub-nodes if all of the plurality of graph
11 observers have been deactivated.

1 13. The method of claim 12, comprising the further step of generating a completed event
2 for each node observer that received an encountered event and that did not cause the
3 graph observer to become deactivated.

1 14. The method of claim 12, comprising the further step of reactivating the graph
2 observer associated with the bound node observer after the set of sub-nodes of the
3 encountered node have been walked.

1 16. The method of claim 12, comprising the further step of walking the sub-nodes if at
2 least one graph observer is active.

1 17. A program product stored on a recordable medium, which when executed, analyzes a
2 graph of hierarchical data, the program product comprising:

3 program code configured to bind a plurality of graph observers to a graph,
4 wherein each graph observer is further bound to a set of node patterns and a set of node
5 observers;

6 program code configured to provide graph walking logic for systematically
7 walking through nodes within the graph;

8 program code configured to provide a first pruning system that can be instructed
9 by a node observer bound with an associated graph observer to deactivate the associated
10 graph observer until a set of sub-nodes for an encountered node has been walked; and

11 program code configured to provide a second pruning system that can instruct the
12 graph walking logic not to walk the set of sub-nodes for the encountered node.

1 18. The program product claim 17, wherein the second pruning system will cause the set
2 of sub-nodes not to be walked only if all of the plurality of graph observers have been
3 deactivated.

1 19. The program product claim 17, further comprising program code configured to
2 provide a pattern testing system for determining if the encountered node matches one of
3 the node patterns.

1 20. The program product claim 17, further comprising program code configured to
2 provide an event manager for generating an encountered event when one of the node
3 observers is bound to a matching node pattern.

A copy of the Prior Art Document is attached hereto as Exhibit A.